

Advanced Natural Language Processing Techniques to Profile Cybercriminals

PROYECTO DE GRADO 1

AUTOR Alejandro ANZOLA ÁVILA

DIRECTOR Daniel Orlando DÍAZ LÓPEZ, *PhD*

22 de mayo de 2019

Escuela Colombiana de Ingeniería Julio Garavito

Agenda

1. Objetivos y justificación
2. Resultados propuestos y productos obtenidos
3. Marco teórico

Clasificador NAÏVE BAYES

Clasificación con SUPPORT VECTOR MACHINES (SVM)

SELF-ORGANIZING MAPS (SOM)

4. Problemas y soluciones

MODELO 1: Predicción de etiquetas de Twitter

MODELO 2: Reconocimiento de NAMED ENTITIES con redes LSTM

MODELO 3: Búsqueda de tweets relacionados con *embeddings*

5. Conclusiones y trabajo futuro

Objetivos y justificación

Objetivo general

Generar herramientas y estrategias para el perfilado de cibercriminales con ayuda de metodologías de *NLP* aplicado a datos recolectados de comunicaciones y redes sociales.

Objetivos específicos

- Diseñar e implementar una solución de lenguaje natural para realizar el perfilado de sospechosos.
- Identificar el estado del arte en sistemas que usan *NLP* para apoyar agencias de seguridad del Estado.
- Implementación de artefactos para la construcción de *datasets* con información recolectada de medios privados como de fuentes abiertas.
- Validar la solución desarrollada frente a un escenario real.
- Modelado de diferentes metodologías, heurísticas y meta-heurísticas para *NLP*.

Por hacer

Resultados propuestos y productos obtenidos

Por hacer

1. Entendimiento de las generalidades de DATA SCIENCE:
 - Tipos de MACHINE LEARNING
 - Sistemas de detección de anomalías
 - Diferentes modalidades de clustering
2. Identificación de modelos de NLP aplicables para el perfilado de cibercriminales
3. Entendimiento de los modelos de clasificación y clustering:
 - Clasificador de Naïve Bayes
 - Maquinas de soporte vectorial
 - Mapas autoorganizados
4. Entendimiento de los modelos utilizados en NLP:
 - Predicción de etiquetas con modelos de regresión lineal
 - Reconocimiento de NAMED ENTITIES

- Uso de *embeddings* generados con STARSPACE para los k textos mas similares
5. Implementación de modelos de NLP para el perfilado de cibercriminales:
- Modelo de predicción de hashtags de Twitter con modelos lineales
 - Modelo de reconocimiento de NAMED ENTITIES con redes LSTM
 - Modelo de clustering en redes SOM con *embeddings* de STARSPACE

Marco teórico

Para variables aleatorias x e y , se tiene que la probabilidad condicional $P(y | x)$ es definida como

$$P(y | x) = \frac{P(x | y)P(y)}{P(x)}$$

Clasificador NAÏVE BAYES

Un clasificador de Naïve Bayes estima la probabilidad condicional de las clases por medio de suponer que los atributos son condicionalmente independientes, dado la etiqueta de clasificación y . Donde cada conjunto de d atributos $\mathbb{X} = \{x_1, \dots, x_d\}$ se tiene

$$P(\mathbb{X} \mid y = y) = \prod_{i=1}^d P(x_i \mid y = y)$$

El clasificador computa la probabilidad posterior para cada clase y como

$$P(y \mid \mathbb{X}) = \frac{P(y) \prod_{i=1}^d P(x_i \mid y)}{P(\mathbb{X})} \Rightarrow P(y) \prod_{i=1}^d P(x_i \mid y)$$

Nota Puede ignorarse $P(\mathbb{X})$ debido a que es un termino constante. Para esto se realiza una normalización con una constante ϵ de forma que $\sum_{y \in \mathbb{Y}} \epsilon^{-1} P(y \mid \mathbb{X}) = 1$.

Clasificación con SUPPORT VECTOR MACHINES (SVM)

Técnica de **clasificación** con una frontera de decisión en forma de hiper-planos que permiten aplicaciones con vectores de alta dimensionalidad.

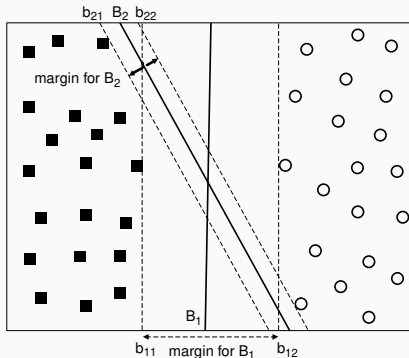


Figura 1: Maximum Margin Hyperplanes. Tomado de [5].

SELF-ORGANIZING MAPS (SOM)

Es un mapa discreto de o neuronas con vectores $\mathbf{w} \in \mathbb{R}^m$ que se adaptan a una entrada de $\mathbf{X} \in \mathbb{R}^{m \times N}$ de N patrones. Tiene una adaptación con una tasa de aprendizaje α_t y un área de afectación σ_t que se reducen por cada iteración $t \in \{0, \dots, T\}$.

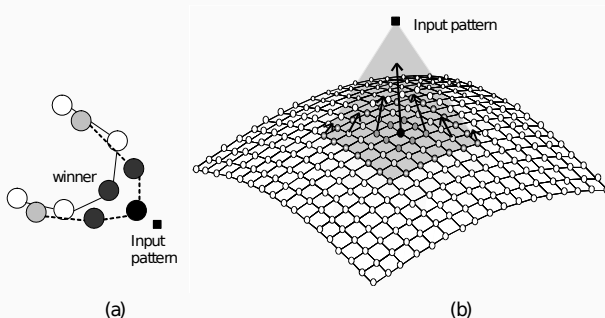


Figura 2: Proceso de adaptación de SOM, (a) uni-dimensional, (b) bi-dimensional. Tomado de [2].

Ejemplo de SOM

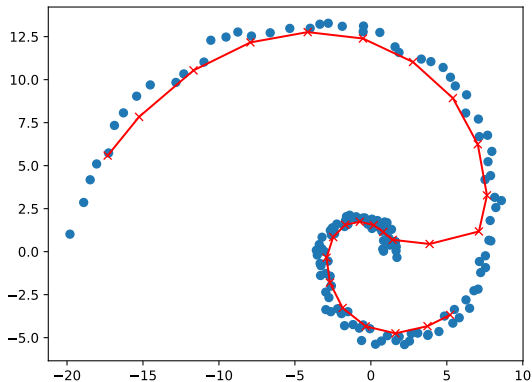


Figura 3: Ejemplo de salida de SOM uni-dimensional con 25 neuronas. Implementación propia.

Aplicación de SOM en perfilamiento de criminales

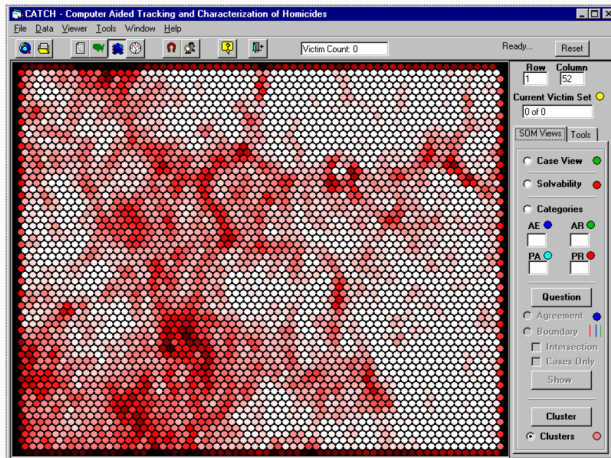


Figura 4: Ejemplo de uso de SOM en aplicaciones de perfilado. Tomado de [4].

Problemas y soluciones

Tweet

*“Really excited to add @plaidavenger to my **#deathlist** along with Italy and @Plaid_Obama after receiving that information. **#KillEveryone #ISIS**”*

MODELO 1: Predicción de etiquetas de Twitter

¿Que hacer?

Con un modelo de regresión lineal predecir los hashtags de los tweets.

“Really excited to add @plaidavenger to my deathlist along with Italy and @Plaid_Obama after receiving that information.” \Rightarrow #deathlist, #KillEveryone, #ISIS

Representación de palabras: BAG OF WORDS

N es el tamaño del diccionario de términos D (e.g. $N = |D|$).

$$\text{word2idx} = \{(t_i, i) : \forall i \in \{1, \dots, N\}\}$$

$$\text{idx2word} = [t_1, \dots, t_N]$$

Representación de palabras en vectores para BoW

Para un término individual su vector representativo se define como:

$$\mathbf{e}^{(i)} = [0, \dots, 1, \dots, 0] \leftarrow \text{posicion } i\text{-ésima}$$

$$\mathbf{e}^{(i)}, (t, i) \in \text{word2idx}$$

Para un documento d de términos, se calcula por cada término que existen dentro del diccionario su vector representativo como:

$$\mathbf{s} = \sum_{(t,i) \in \text{word2idx}} \mathbf{e}^{(i)}, t \in d$$

Representación de palabras: TF-IDF

TF-IDF = Term Frequency – Inverse Document Frequency

Propósito

Darle mayor importancia a las palabras que ocurren con frecuencia intermedia en el documento d y en el corpus D .

$\text{tf}(t, d)$ = Frecuencia del termino (o n-grama) t en el documento d

$$\text{idf}(t, D) = \log\left(\frac{N}{|\{d \in D : t \in d\}|}\right); N = |D|$$

$$\text{tf-idf}(t, d, D) = \text{tf}(t, d) \cdot \text{idf}(t, D)$$

Regresión lineal

Para un vector de parámetros $\boldsymbol{\theta}$ y un vector de características \mathbf{x} , la regresión lineal se puede definir como:

$$\hat{y}(\mathbf{x}, \boldsymbol{\theta}) = \boldsymbol{\theta}^\top \mathbf{x} = \theta_0 + \theta_1 x_1 + \cdots + \theta_n x_n$$

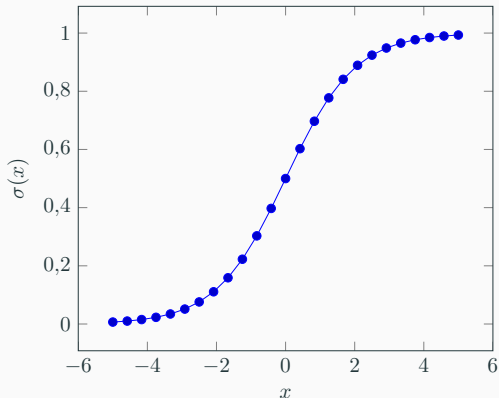
Donde $\hat{y}(\mathbf{x}, \boldsymbol{\theta}) : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$.

θ_0 se le conoce como el *bias* del modelo.

El objetivo es que para una salida esperada y se tenga la salida \hat{y} con menor error por medio de ajustar los valores de $\boldsymbol{\theta}$. De forma que se quiere:

$$\boldsymbol{\theta} = \arg \min_{\boldsymbol{\theta}} |\hat{y}(\mathbf{x}, \boldsymbol{\theta}) - y|$$

Regresión logística $\sigma(x)$



$$\sigma(x) = \frac{1}{1 + \exp(-x)}$$

$$\sigma(x) : \mathbb{R} \rightarrow (0, 1)$$

Evita problemas de BIAS
y OVERFITTING del modelo

Figura 5: Gráfica de función sigmoide.

One vs Rest

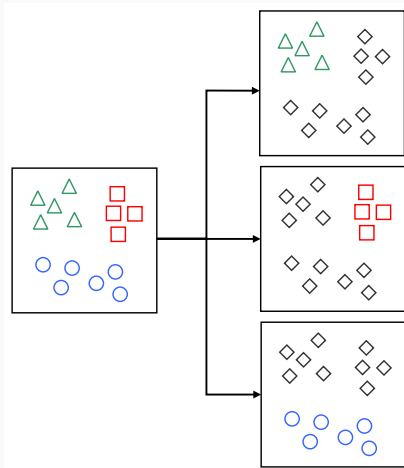


Figura 6: Algoritmo de One vs Rest.

Se entrenan C estimadores θ_i para cada clase con algún algoritmo de optimización (ej. gradiente descendiente).

Se determina un estimador $c \in \{1, \dots, C\}$, que se calcula como:

$$c = \arg \max_i \sigma(\theta_i^\top \mathbf{x})$$

A partir de un diccionario previamente definido a entrenar el ONE vs REST de forma:

$$\{(i, h)\}; h \in \text{hashtags}; i \in \{1, \dots, C\}$$

De forma que se recupera el hashtag h correspondiente a partir de la clase estimada i por ONE vs REST.

¿De que y de quienes están hablando?

Tweet: @realDonaldTrump

*“The **Democrats** new and pathetically untrue sound bite is that we are in a “Constitutional Crisis.” They and their partner, the **Fake News Media**, are all told to say this as loud and as often as possible. They are a sad JOKE! We may have the strongest **Economy** in our history, best ...”*

MODELO 2: Reconocimiento de NAMED ENTITIES con redes LSTM

Son redes neuronales recurrentes que son capaces de reconocer NAMED ENTITIES.

Texto	Donald	Trump	es	presidente	de	Estados	Unidos
Etiqueta	B-PER	I-PER	O	O	O	B-ORG	I-ORG

Cuadro 1: Ejemplo de reconocimiento de NAMED ENTITIES.

Otro	O
Persona	PER
Ubicación	LOC
Organización	ORG
Misceláneo	MISC

Cuadro 2: Categorías de NAMED ENTITIES.

Redes neuronales recurrentes (RNN)

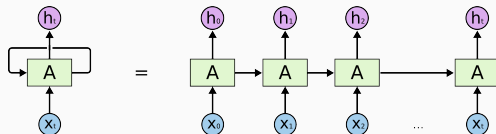


Figura 7: Red RNN simplificada. Tomado de [1].

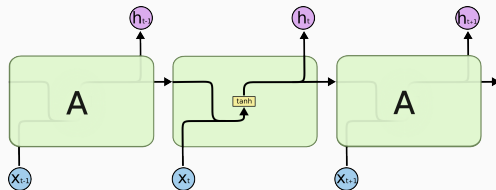


Figura 8: Arquitectura RNN clásica. Tomado de [1].

Redes LONG SHORT TERM MEMORY (LSTM)

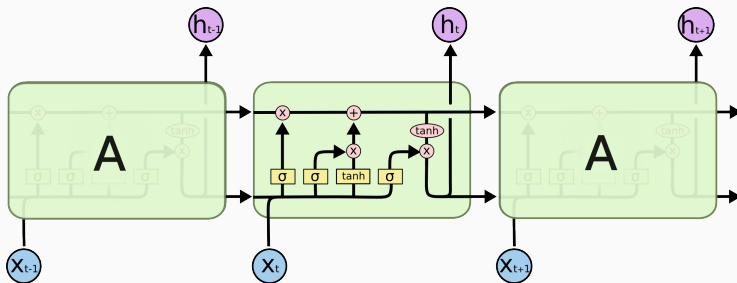


Figura 9: Arquitectura de red LSTM clásica. Tomado de [1].

Nota

Estas redes son solo *feedforward* (e.g. hacia adelante). Solo se basan en entradas pasadas.

Redes BIDIRECTIONAL LONG SHORT TERM MEMORY (Bi-LSTM)

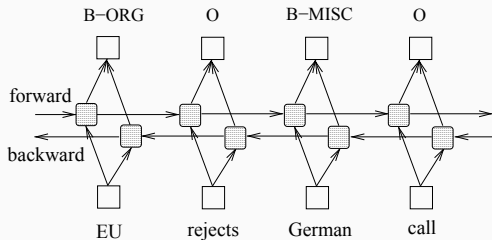


Figura 10: Etiquetado con una Bi-LSTM. Tomado de [3].

Nota

Estas redes son *feedforward* como *backward*. Se basan de entradas pasadas y futuras.

Se procede primeramente por entrenar el modelo con un dataset, que establece la relación (t, e) de las palabras t y las *Entities* e .

Las librerías en uso son TENSORFLOW y NUMPY.

¿Que son los *embeddings*?

Son espacios de vectores n -dimensionales que se mapean según una palabra.

Tómese \mathbf{p}_t como el vector que representa el termino t y a d como la distancia calculada entre los vectores (típicamente la distancia **coseno**).

Ejemplo

$d(\mathbf{p}_{\text{asombroso}}, \mathbf{p}_{\text{genial}})$ debería tener un valor bajo.

$d(\mathbf{p}_{\text{asombroso}}, \mathbf{p}_{\text{terrible}})$ debería tener un valor alto.

También se pueden representar varias palabras de un documento en un solo vector por medio de sumarlos. (i.e. $\sum_{t \in d} \mathbf{p}_t$).

MODELO 3: Búsqueda de tweets relacionados con *embeddings*

Es posible categorizar los k textos mas parecidos a una consulta q en base a su embedding con otros textos recopilados.

StarSpace

Genera embeddings en base a un dataset de entrenamiento.
Desarrollado por Facebook Research en 2017 [6].

Conclusiones y trabajo futuro

Por hacer

- Implementación de los modelos 2, 3, 4 propuestos
- Implementar un modelo de recolección de información de redes sociales de ciberdelinquentes de forma que sea mas fácil perfilarlos contra futuros
- Realizar una visualización en dashboard de algoritmos de clustering como SOM

- [1] Understanding lstm networks.
- [2] L. N. De Castro.
Fundamentals of natural computing: basic concepts, algorithms, and applications.
Chapman and Hall/CRC, 2006.
- [3] Z. Huang, W. Xu, and K. Yu.
Bidirectional LSTM-CRF Models for Sequence Tagging.
2015.
- [4] J. Mena.
Investigative Data Mining for Security and Criminal Detection.
Elsevier Science, 2003.

- [5] P.-N. Tan, M. Steinbach, and V. Kumar.
Introduction to Data Mining.
Addison Wesley, us ed edition, May 2005.
- [6] L. Wu, A. Fisch, S. Chopra, K. Adams, A. Bordes, and J. Weston.
Starspace: Embed all the things!
arXiv preprint arXiv:1709.03856, 2017.